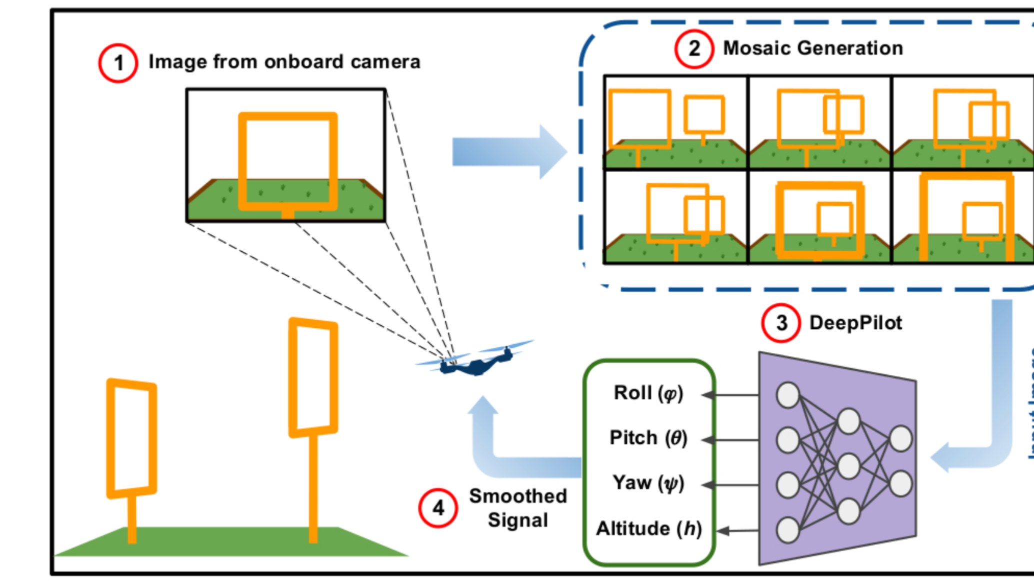
DGMD E-17 Final Project

Summary Statement

CNN approach called that takes onboard camera images as input and predicts 4 flight commands as output.

evaluations are carried out in simulation using the Gazebo simulator in simple tracks with gates always parallel to the drone's camera plane.



Note - traditional pose approach

inspired by 6-DoF camera pose (rotation and translation) prediction from a single camera using a CNN architecture called PoseNet, but is aimed at inferring the drone's flight commands rather than a 6-DoF pose

Work based on

Input

Output

Terminology

Main Steps

Process

Neural Network Model

Architecture

Drone Model

Quadcopter Model

To model the drone used in simulations, use a quadcopter model with 6 Degrees of Freedom (DoF)

provided by the Gazebo simulator

available with the Robotic Operating System (ROS)

This model is based on the AR.Drone model offered by the Parrot Company

The model expects 4 flight commands (ϕ, θ, ψ, h)

Note that the flight commands (ϕ, θ, ψ, h) are expected to have values in the range [−1, 1]

Degrees of freedom

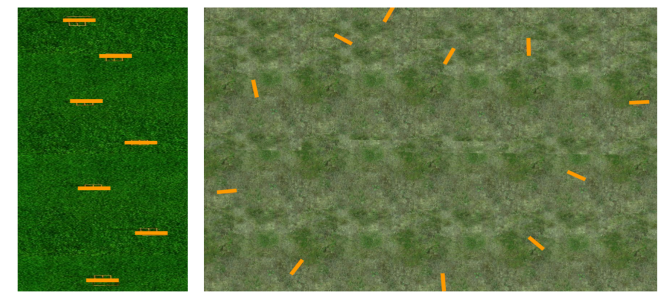
- They have six degrees of freedom, three translational and three rotational, but only four independent inputs (rotor speeds), thus creating an underactuated system. Translational and rotational motion is couples in order to achieve six degrees of freedom.
- The quadcopter has 6 degree of freedom. This means that 6 variables are needed to express its position and orientation in space (x, y, z, ϕ, θ and ψ).
- The drone can move with six degrees of freedom, meaning it can not only move horizontally and vertically along the x,y and z axes, it can also rotate between those axes.

at the time of the research, the goal was to test the concept with the physical vehicle, and one choice is to use the Bebop 2.0 Power Edition (PE). A drone that uses the same flight commands, with a speed of 16m/s and with a Software Development Kit that enables software implementation on ROS

Need to see what physical model is applicable at this time for the 4 flight commands

Data Acquition

to create image dataset, manually flew drone in two racetracks in Gazebo



a. This racetrack is composed of 7 gates 2 m in height. The track spans over a surface of 53.5 m × 9.6 m, and space in between gates from 10 m to 12 m.

b. A second racetrack composed of 3 gates 3.5 m in height, 4 gates 2 m in height and 4 gates 1.2 m in height, randomly positioned. The track spans over a surface of 72 m × 81 m, and space in between gates from 2 m to 12 m

acquired images to teach the model 7 basic movements also considering gates with variations in height and orientation

For each one of these images, a tuple of flight commands (ϕ, θ, ψ, h) is also recorded. These flight commands will be used as labels for the training stage, and as ground truth of the validation dataset.

Lateral, forward, rotational, and vertical.

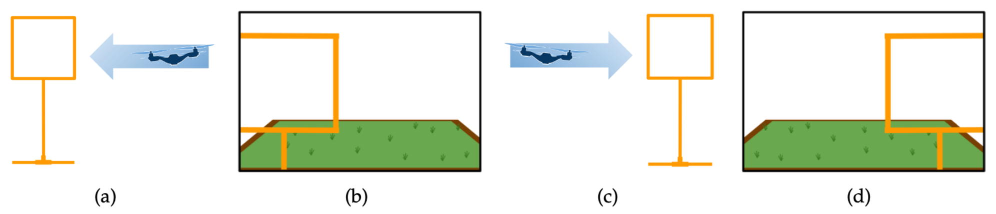


Figure 5. Schematic drone's lateral motion: **(a)** outside view of the corresponding side motion when the gate appears to the left of the image **(b)**; **(c)** corresponding side motion when the gate appears to the right of the image **(d)**. The flight command ϕ will take values in the range of $[-1, 1]$; values for (θ, φ, h) will be set to zero.

Lateral

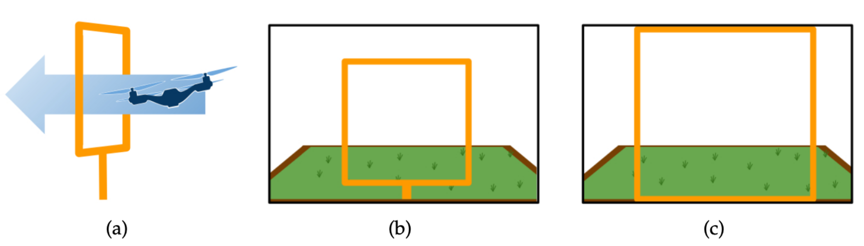


Figure 6. Schematic drone's forward motion: **(a)** outside view of the corresponding forward motion when the gate appears in the image center **(b,c)**. The flight command θ will take values in the range of $[0, 1]$; values for (ϕ, φ, h) will be set to zero.

Forward

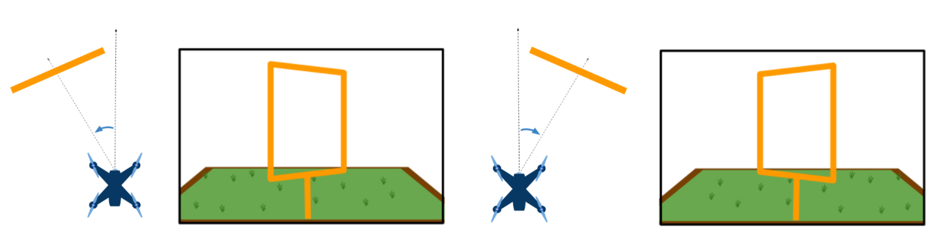


Figure 7. Schematic drone's rotational motion: **(a)** outside view of the corresponding rotational motion in the yaw angle when the gate appears skewed towards the right of the image **(b)**; **(c)** corresponding yaw motion when the gate appears skewed towards the left of the image **(d)**. The flight command ψ will take values in the range of $[-1, 1]$; values for (ϕ, θ, h) will be set to zero.

Yaw

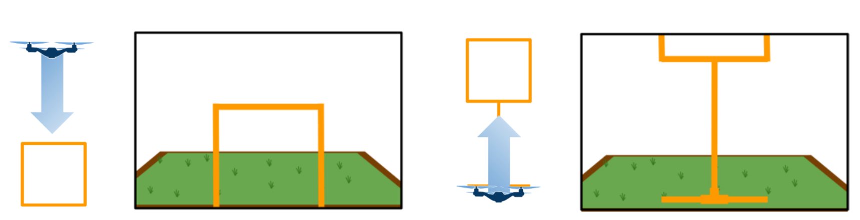


Figure 8. Schematic drone's vertical motion: **(a)** outside view of the drone's motion flying upwards when the gate appears at the bottom of the image **(b)**; **(c)** downwards motion when the gate appears at the top of the image **(d)**. The flight command h will take values in the range of $[-1, 1]$; values for (ϕ, θ, ψ) will be set to zero.

Altitude (Vertical)

In the collected images, the gate was always kept in the camera's viewing area.

For each image, the flight commands given by the pilot are saved as the tuple (ϕ, θ, ψ, h).

MANUAL ADJUSTMENT

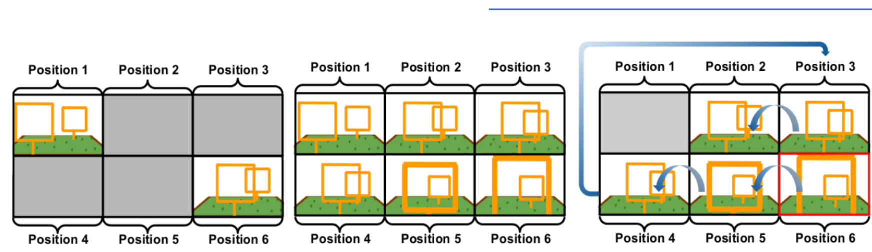
after the recording, a manual adjustment is made to the flight commands to identify the dominant drone's motion as illustrated in the schematic figures. Once identified, the flight command corresponding to the dominant motion will be kept as recorded, whereas the other values in the tuple are set to zero.

This was made to avoid ambiguous flight commands, which ultimately would be used as labels to our datasets.

captured a total of 14,900 images with their respective flight commands (ϕ, θ, ψ, h) for each image, while the pilot controlled the drone manually.

13,050 images for training and 1850 for validation

14,900 images, a total of 41,360 mosaic images were generated (across the 4 dataset of 2,4,6,8 image mosaics)



To compose the mosaic, the spaces in the mosaic image are filled every 5 frames:

Once the mosaic is full, the first frame in the mosaic is removed from position 1, and all frames are shifted to the left, this is, the frame in the position 2 moves to the position 1, and the frame in the position 3 moves to the position 2 and so on.

note that the image in position 4 moves to position 3 when the shift occurs

set of racetrack worlds in Gazebo

range sensors

obstacles